



INSTITUTO TECNOLÓGICO SUPERIOR DE JEREZ

**Jerez, Zac
01 de mayo del 2019**

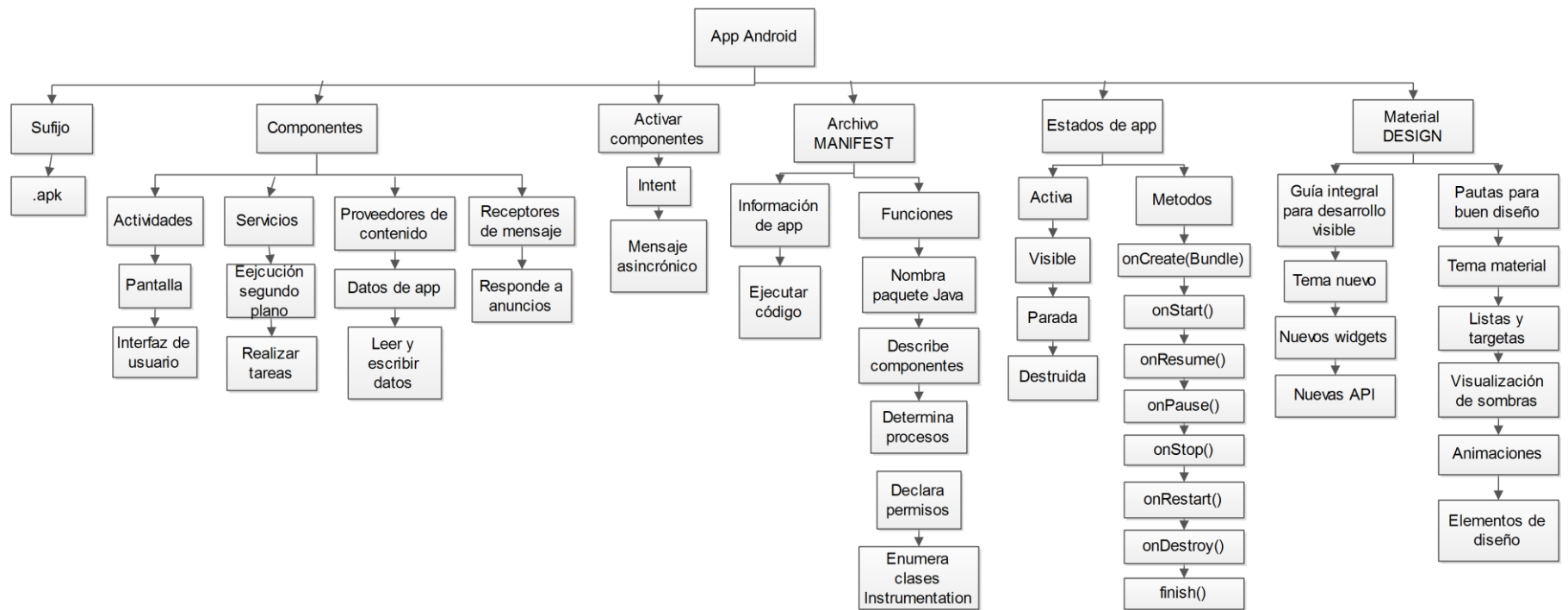
**Alumna: Leticia Carrera Venegas
Núm. control: S17070155
Correo electrónico: Letycv25@gmail.com**

**Ingeniería en Sistemas Computacionales
Semestre: 4**

**Tópicos avanzados de programación
Actividad: mapa conceptual**

Docente: I.S.C. Salvador Acevedo Sandoval.





1. ¿Cuál es el sufijo para las aplicaciones que se instalan en Android?

Apk

2. ¿Cuáles son los 4 componentes que forman a una aplicación Android?

Actividades: representa una pantalla con interfaz de usuario.

Servicios: es un componente que se ejecuta en segundo plano para realizar operaciones prolongadas o tareas para procesos remotos. Un servicio no proporciona una interfaz de usuario.

Proveedores de contenido: administra un conjunto compartido de datos de la app, también son útiles para leer y escribir datos privados de tu aplicación y que no se comparten.

Receptores de mensaje: es un componente que responde a los anuncios de mensajes en todo el sistema.

3. ¿Cómo se "activan" dichos componentes?

Se activan mediante un mensaje asincrónico llamado *intent*.

Una intent se crea con un objeto `Intent`, que define un mensaje para activar un componente específico o un *tipo* específico de componente; una intent puede ser explícita o implícita, respectivamente.

4. ¿Qué es el archivo MANIFEST y para qué sirve?

Proporciona información esencial sobre tu aplicación al sistema Android, información que el sistema debe tener para poder ejecutar el código de la app.

- Nombra el paquete de Java para la aplicación. El nombre del paquete sirve como un identificador único para la aplicación.
- Describe los componentes de la aplicación, como las actividades, los servicios, los receptores de mensajes y los proveedores de contenido que la integran. También nombra las clases que implementa cada uno de los componentes y publica sus capacidades, como los mensajes `Intent` con los que pueden funcionar. Estas declaraciones notifican al sistema Android los componentes y las condiciones para el lanzamiento.
- Determina los procesos que alojan a los componentes de la aplicación.

- Declara los permisos que debe tener la aplicación para acceder a las partes protegidas de una API e interactuar con otras aplicaciones. También declara los permisos que otros deben tener para interactuar con los componentes de la aplicación.
- Enumera las clases Instrumentation que proporcionan un perfil y otra información mientras la aplicación se ejecuta. Estas declaraciones están en el manifiesto solo mientras la aplicación se desarrolla y se quitan antes de la publicación de esta.
- Declara el nivel mínimo de Android API que requiere la aplicación.
- Enumera las bibliotecas con las que debe estar vinculada la aplicación.

5. ¿Cuáles son los estados en los que se puede encontrar una app?

Activa (Running): La actividad está encima de la pila, lo que quiere decir que es visible y tiene el foco.

Visible (Paused): La actividad es visible pero no tiene el foco. Se alcanza este estado cuando pasa a activa otra actividad con alguna parte transparente o que no ocupa toda la pantalla. Cuando una actividad está tapada por completo, pasa a estar parada.

Parada (Stopped): Cuando la actividad no es visible. El programador debe guardar el estado de la interfaz de usuario, preferencias, etc.

Destruída (Destroyed): Cuando la actividad termina al invocarse el método finish(), o es matada por el sistema.

6. ¿Cuáles son los métodos que permiten manipular dichos estados?

onCreate(Bundle): Se llama en la creación de la actividad. Se utiliza para realizar todo tipo de inicializaciones, como la creación de la interfaz de usuario o la inicialización de estructuras de datos. Puede recibir información de estado de la actividad (en una instancia de la clase Bundle), por si se reanuda desde una actividad que ha sido destruida y vuelta a crear.

onStart(): Nos indica que la actividad está a punto de ser mostrada al usuario.

`onResume()`: Se llama cuando la actividad va a comenzar a interactuar con el usuario. Es un buen lugar para lanzar las animaciones y la música.

`onPause()`: Indica que la actividad está a punto de ser lanzada a segundo plano, normalmente porque otra actividad es lanzada. Es el lugar adecuado para detener animaciones, música o almacenar los datos que estaban en edición.

`onStop()`: La actividad ya no va a ser visible para el usuario. Si hay muy poca memoria es posible que la actividad se destruya sin llamar a este método.

`onRestart()`: Indica que la actividad va a volver a ser representada después de haber pasado por `onStop()`.

`onDestroy()`: Se llama antes de que la actividad sea totalmente destruida. Por ejemplo, cuando el usuario pulsa el botón de volver o cuando se llama al método `finish()`. Si hay muy poca memoria, es posible que la actividad se destruya sin llamar a este método.

7. ¿Qué es y para qué sirve MATERIAL DESIGN?

Material Design es una guía integral para el diseño visual, de movimientos y de interacción en distintas plataformas y dispositivos

- un tema nuevo;
- nuevos widgets para vistas complejas;
- nuevas API (interfaces de programación de aplicaciones) para sombras y animaciones personalizadas.

8. ¿Cuáles son las 6 grandes pautas que especifica MATERIAL DESIGN para un buen diseño de apps?

Tema material: configurar paleta de colores y animaciones predeterminadas para información táctil y transiciones de actividades.

Listas y tarjetas: Android proporciona dos widgets nuevos para mostrar listas y tarjetas con estilos y animaciones de Material Design:

El nuevo widget RecyclerView es una versión más acoplable de ListView que admite diferentes tipos de diseños y proporciona mejoras en el rendimiento.

El nuevo widget CardView te permite mostrar extractos de información importante dentro de tarjetas que tienen apariencia y estilo coherentes.

Visualización de sombras

Además de las propiedades X e Y, las vistas de Android ahora poseen una propiedad Z. Esta propiedad nueva representa la elevación de una vista, que determina lo siguiente:

- El tamaño de la sombra: las vistas con valores Z más elevados proyectan sombras más grandes.
- El orden del dibujo: las vistas con valores Z más elevados aparecen sobre las otras vistas.

Animaciones: información táctil, efecto circular, transición de actividades, movimiento curvo, cambio de estados de la vista.

Elementos de diseño: dibujables en vector, teñido de los dibujables y extracción de color.

Referencias

Developers, A. (01 de 05 de 19). *Aspectos fundamentales de la aplicación*. Obtenido de <https://developer.android.com/guide/components/fundamentals?hl=es-419>

Developers, A. (30 de 04 de 19). *Manifiesto de la app*. Obtenido de <https://developer.android.com/guide/topics/manifest/manifest-intro?hl=es-419>

Developers, A. (01 de 05 de 19). *Material Design para Android*. Obtenido de <https://developer.android.com/design/material?hl=es-419>

UPV. (01 de 05 de 19). *Ciclo de vida de una actividad*. Obtenido de <http://www.androidcurso.com/index.php/tutoriales-android/37-unidad-6-multimedia-y-ciclo-de-vida/158-ciclo-de-vida-de-una-actividad>